MA 3232 - Numerical Analysis
Final Exam - Quarter II - AY 04-05

Instructions: Work all problems. Read the problems carefully. Show appropriate work, as partial credit will be given. Three pages ($8\frac{1}{2}$ by 11) of notes (both sides) and Mathematics Department "Blue Books" of tables permitted.

---

1. (30 points) Consider the table of data:

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$
0.00	3.000	1.633	−1.946
0.35	4.633	−0.3130	−0.3670
0.70	4.320	−0.6800	0.1850
1.05	3.640	−0.4950	0.2560
1.40	3.145	−0.2390	
1.75	2.906		

a. Estimate $f(0.83)$ using the *most appropriate* second-degree Newton-Gregory forward polynomial.

solution:

The partially complete difference table, up through the third differences, is:

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$
0.00	3.000	1.633	−1.946	1.579
0.35	4.633	−0.3130	−0.3670	0.5520
0.70	4.320	−0.6800	0.1850	0.0710
1.05	3.640	−0.4950	0.2560	
1.40	3.145	−0.2390		
1.75	2.906			

By definition, the second-degree Newton-Gregory forward difference polynomial has the form:

$$P_2(x) = f_0 + s\ \Delta f_0 + \frac{s(s-1)}{2}\Delta^2 f_0$$

solution:

In this case, since we need a quadratic, we have to use three points. The most appropriate choice is the closest three, which in this case means:

$$0.35,\ 0.70\ \text{ and }\ 1.05 \qquad \Longrightarrow \qquad x_0 = 0.35$$

Also, then, for $x = 0.83$,

$$s = \frac{x - x_0}{h} = \frac{0.83 - 0.35}{0.30} = 1.371$$

Therefore, using the line in the difference table that starts at $x_0 = 0.35$, we have

$$\begin{aligned} P_2(0.83) &= (4.633) + (1.371)(-0.3130) + \frac{(1.371)(1.371-1)}{2}\,(-0.3670) \\ &= 4.111 \end{aligned}$$

(Note for the usually less appropriate choice, $x_0 = 0.70$, we would get: $f(0.83) \doteq 4.046$.)

b. Estimate the error in your answer to part a.

solution:

The normal error estimate in a Newton-Gregory interpolation is the "next term," i.e., in this case

$$\begin{aligned} \frac{s*(s-1)*(s-2)}{6}\,\Delta^3 f_0 &= \frac{(1.371)*(1.371-1)*(1.371-2)}{6}\,(0.5520) \\ &\doteq -0.0294 \end{aligned}$$

which would indicate our answer in part a. should be accurate to two about significant digits. (Note had we choosen to use $x_0 = 0.70$, our estimate error would have been 0.0045.)

solution:

Note the actual function we used here was

$$f(x) = 10xe^{-2x}\cos(x) + 3$$

and the actual value is:

$$f(0.83) \doteq 4.065$$

which implies that the actual error using $x_0 = 0.83$ is

$$E = -0.046$$

This value is clearly the same order of magnitude as our estimate. Note that had we instead used $x_0 = 0.70$ we would have obtained an actual error of

$$E = 0.019$$

which is not accurately estimated by the "next term" estimate in that case of 0.0045.

c. What would have been the advantages and disadvantages of using a cubic spline here instead of second-degree Newton-Gregory polynomial?

solution:

There would probably have been two advantages to using a cubic spline. First of all, cubics generally produce more accurate estimates than quadratics. (That in fact is the case here. MATLAB's **interp1( )** spline returns a value of $f(0.83) \doteq 4.062$, which is significant more accurate than either Newton-Gregory value in part a. Secondly, if we needed to graph the function here, a spline fit would generally produce a more pleasant-appearing graph.

The major drawbacks to using a cubic spline would be that, first, the equations for fitting a spline are much more difficult (we didn't even cover them), and involve solving a system of coupled linear equations, i.e. a matrix. Secondly, we have not developed any methods to estimate the error of a spline fit, and consequently would have no way to tell how much confidence we could have in our answer.

2. (30 points) Consider the function:

$$f(x) = x^3 - 4x + 1 \quad .$$

a. Show that this function has a *simple* root in the interval $0 < x < 1$.

solution:

First, observe that

$$f(0) = 1 \ \text{ and } \ f(1) = -2$$

The sign change immediately implies that at least one root exists in the interval. But also, note that

$$f'(x) = 3x^2 - 4 < 0 \ \text{ for } \ 0 \le x \le 1$$

and therefore, $f'(r) \neq 0$ at the root. Hence the root is simple.

b. Estimate this root using *two* iterations of the *Secant Method.*

solution:

The secant algorithm is:

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$$

Since the residuals at the left hand end is smaller, and hence likely the root is closer to that end, we will choose

$$x_0 = 0 \qquad \Longrightarrow \qquad x_{-1} = 1$$

This immediately implies

$$x_1 = x_0 - \frac{f(x_0)(x_0 - x_{-1})}{f(x_0) - f(x_{-1})} = (0) - \frac{(1)((0) - (1))}{(1) - (-2)} = \frac{1}{3}$$

solution:

Noting that $f(x_1) = f(1/3) = -0.2962$, we continue

$$x_2 = x_1 - \frac{f(x_1)(x_1 - x_0)}{f(x_1) - f(x_0)}$$
$$= (0.3333) - \frac{(-0.2962)((0.3333) - (0.0000))}{(-0.2962) - (1.000)} = 0.2571$$

(Note that had we chosen $x_{-1} = 0$ and $x_0 = 1$, a generally less than optional choice, we would have obtained:

$$x_1 = .3333 \quad \text{and} \quad x_2 = .2174 \quad)$$

c. Estimate the error in your answer to part b.

solution:

The error estimate in any iterative method (such as secant) is simply the difference between the current iterate and the "next term." In this case, since $f(x_2) = f(0.2571) = -0.0114$, the next term is:

$$x_3 = x_2 - \frac{f(x_2)(x_2 - x_1)}{f(x_2) - f(x_1)}$$
$$= (0.2571) - \frac{(-0.0114)((0.2571) - (0.3333))}{(-0.0114) - (-0.2963)} = 0.2540$$

Therefore, the estimated error in x_2 is

$$e_2 \doteq x_3 - x_2 = 0.2540 - 0.2571 = -0.0031$$

(Note the "exact" answer, from MATLAB's **fzero()** function, is

$$r = 0.254101688...$$

and therefore the actual error is -0.0030, i.e. our estimate is right on.

solution:

Note also, had we started with the gnerally less than optimal choice of

$$x_{-1} = 0 \quad \text{and} \quad x_0 = 1$$

we would have obtained

$$x_3 = .2547 \qquad \Longrightarrow \qquad e_2 \doteq .2547 - .2174 = 0.0373 \ ,$$

i.e. almost an order of magnitude greater than the error with the more preferable choice.

d. Would *Newton's method* have been preferable in **this** problem? (*Briefly* explain your answer!)

solution:

The key here is to first clearly indicate the criteria by which "preferable" will be judged. The usual ones are efficiency, measured in terms of function evalauations required, and effectiveness, measured in terms of accuracy of the iterate. Newton then will be preferable if it produces a more accurate solution with less work.

Under these criteria, the answer here is almost certainly probably not! In general, we know secant converges fairly quickly, probably picking up more than one digit per iteration. (We know that Newton, which is only slightly faster, usally picks up two digits per iteration.) Moreover, there is nothing in the behavior of our first couple of iterates to suggest that would not also be the case here. Therefore, assuming our error estimate accurately represents the order of magnitude of the errors, we would expect that one more iteration of the secant method, i.e. computing x_4, would produce an answer with an error on the order of magnitude of 0.0005 or smaller, i.e. four-digit machine precision. This would be for a total cost of four function evaluations, i.e. about the same as we would need for only two iterations of Newton. There is very little chance two iterations of Newton would have produced a significantly better answer than our x_3. (Actually, it does only about as well, so it's not clearly preferable.)

However, note that with the less optimal choice of starting values (i.e. $x_0 = 1$) and the consequent significantly larger estimate to e_2, then the likelihood that sequent might require at least two more iterations to converge increases, and Newton becomes potentially more attractive.

3. (40 points) Consider the following table of data:

x	$f(x)$
0.00	0.0000
0.25	0.6065
0.50	0.7358
0.75	0.6694
1.00	0.5413
1.25	0.4104
1.50	0.2987

a. Approximate $\int_0^{1.5} f(x)dx$ using *Simpson's* rule and a step size of $h = 0.75$

solution:

Note that, in this case, Simpson's rule with $h = 0.75$ would utilize only three data points. Therefore, we must use the local form, i.e.

$$\int_{x_i}^{x_{i+2}} f(x)\,dx \doteq \frac{h}{3}\Big[f_i + 4f_{i+1} + f_{i+2}\Big]$$

or, for this set of data

$$\int_0^{1.5} f(x)\,dx \doteq \frac{0.75}{3}\Big[(0.0000) + 4(0.6694) + (0.2987)\Big] = 0.7441$$

b. Approximate $\int_0^{1.5} f(x)dx$ using the *Trapezoidal* rule and a step size of $h = 0.25$

solution:

The (global) Trapezoidal rule is

$$\int_a^b f(x)\,dx \doteq \frac{h}{2}\Big[f_0 + 2f_1 + 2f_2 + \cdots + 2f_{N-1} + f_N\Big]$$

solution:

or, for this set of data, with $h = 0.25$,

$$\int_0^{1.5} f(x)\,dx \doteq \frac{0.25}{2}\Big[(0.0000) + 2(0.6065) + 2(0.7358) + 2(0.6694) + 2(0.5413) + 2(0.4104) + (0.2987)\Big] = 0.7782$$

c. Can you use your answer to part a. to estimate the error in your answer to part b? (*Briefly* explain your answer.) If not, briefly describe what method you would use. (In the latter case, **do not** actually do the calculations!)

solution:

Estimating the error in an integration rule requires either:

(i.) A "next term" estimate, produced by comparing results computed using two different order methods but the *same step size* (h), or

(ii.) An extrapolated estimated, produced by comparing results computed using the *same method*, but with two different step sizes.

In this example, we have two estimates computed using *different methods* and *different step sizes*. Therefore, neither of the above paradigms apply.

To obtain an error estimate here for the answer in part b, we would first have to compute either:

(i.) A Simpson's rule result using $h = 0.25$, and then apply the "next term" estimate, or

(ii.) A Trapezoidal rule result using $h = 0.50$, and then apply Romberg extrapolation.

d. Would three-point Gaussian quadrature have been an appropriate method for solving this problem? (*Briefly* explain your answer!)

solution:

Gaussian quadrature generally produces significantly more accurate answers that Newton-Cotes, but to do so, it must use *unevenly space* data points, where the spacing must be chosen to precisely match the order of the estimate. That is easily done when we know the analytic form of the function. However, in this case we are not given that information, and have only the evenly-space table of data above. Therefore, unless we are willing to interpolate these values at the Gauss quadrature points (a step that will introduce further error), we cannot even calculate the Gauss quadrature answer for this table of data. Moreover, even if we could, we have not developed any formulas for estimating the error using Gaussain quadrature, and so could not answer part b. at all. Therefore, we Guassian quadrature does not appear to be appropriate for this data.

4. (30 points) Consider the boundary value problem

$$\begin{aligned} y'' + xy' + y &= x^2 \\ y(0) &= 3 \\ y(2) &= 0 \end{aligned}$$

This can solved approximately, using either Finite Difference or Residual-Based Methods.

a. Write (but **do not solve**), as explicitly as possible, the equations that would result if this problem were solved by second-order, centered finite differences with a step size $h = 1/2$.

solution:

To use finite differences, we must first divide the interval up into a grid, using, in this case, $h = 0.5$. This produces

$$\underbrace{x_0 = 0}_{\text{boundary}} \ , \ \underbrace{x_1 = 0.5 \ , \ x_2 = 1.0 \ , \ x_3 = 1.5}_{\text{interior}} \quad \text{and} \quad \underbrace{x_4 = 2.0}_{\text{boundary}}$$

At each interior point (i.e. at i=1,2,3) we replace *all* of the derivatives in the differential equation by corresponding second-order centered differences, i.e.

$$y'' \ + \ xy' \ + \ y \ = \ x^2$$

becomes

$$\frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} + x_i \ \frac{-y_{i-1} + y_{i+1}}{2h} + y_i = x_i^2$$

or

$$\Big(2 - x_i h\Big) y_{i-1} + \Big(-4 + 2h^2\Big) y_i + \Big(2 + x_i h\Big) y_{i+1} = 2x_i^2 h^2 \ , \quad i = 1, 2, 3$$

More specifically, at $i = 1$ ($x_i = 0.5$) we would have

$$1.75 y_{i+1} - 3.50 y_i + 2.25 y_{i-1} = .1250$$

while at $i = 2$ ($x_i = 1.0$) we would have

$$1.5 y_{i+1} - 3.50 y_i + 2.5 y_{i-1} = .5000$$

and at $i = 3$ ($x_i = 1.5$) we would have

$$1.25 y_{i+1} - 3.50 y_i + 2.75 y_{i-1} = 1.125$$

solution:

Finally, at the boundaries, we would have

$$y_0 = 3 \quad \text{and} \quad y_4 = 0$$

Putting all this together, we have

$$\begin{array}{llllll} y_0 & & & & & = 3 \\ 1.75y_0 & -\ 3.50y_1 & +\ 2.25y_2 & & & = .1250 \\ & 1.50y_1 & -\ 3.50y_2 & +\ 2.50y_3 & & = .5000 \\ & & 1.25y_2 & -\ 3.50y_3 & +\ 2.75y_4 & = 1.125 \\ & & & & y_4 & = 0 \end{array}$$

or, in matrix-vector form

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1.75 & -3.50 & 2.25 & 0 & 0 \\ 0 & 1.50 & -3.50 & 2.50 & 0 \\ 0 & 0 & 1.25 & -3.50 & 2.75 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 3.000 \\ .1250 \\ .5000 \\ 1.125 \\ 0.000 \end{bmatrix}$$

b. Write, as explicitly as possible, the form of the trial functions that should be used if this system were going to be solved using the Galerkin Method and fourth-degree polynomials. (You do **not** need to find the resulting system of equations, only the trial functions.)

solution:

The first key point in creating trial (or test) functions is that *they must satisfy the boundary conditions.* This is most easily done by properly selecting the coefficients of the linear terms, i.e. picking c_0 and c_1 such that

$$c_0 + c_1 x$$

satisfies

$$y(0) = 3 \quad \text{and} \quad y(2) = 0$$

This easily leads to

$$c_0 = 3 \quad \text{and} \quad c_1 = -\frac{3}{2}$$

solution:

The second key point is that adding subsequent terms must not mess up these conditions, i.e., in this case, any subsequent terms must be identically zero at both ends. If we wish to do this with polynomials up to and including degree four, that makes the arguably simplest form for the trial functions

$$u(x) = 3 - \frac{3}{2}\,x + c_2 x(x-2) + c_3 x^2(x-2) + c_4 x^3(x-2)$$

5. (45 points) Consider the initial value problem

$$y' = x + y$$
$$y(0) = 0.$$

(Note the exact solution to this is $y(x) = e^x - x - 1$.)

The *second order Improved (or Modified) Euler* method, used with a step size (h) of 0.25 and only a single correction per step, yields the following table of data

x_i	y_i	f_i
0.00	0.0000	0.0000
0.25	0.0312	0.2812
0.50	0.1416	0.6416
0.75	0.3533	1.1033
1.00	0.6949	

a. Estimate $y(1.0)$ again, this time using only two steps of the *same* method.

solution:

The modified Euler algorithm is

$$y_{n+1,p} = y_n + hf(x_n, y_n)$$
$$y_{n+1,c} = y_n + \frac{h}{2}\Big(f(x_n, y_n) + f(x_{n+1}, y_{n+1,p})\Big)$$

For the differential equation, if we need to find $y(1)$ in two steps, we must use $h = 0.5$. So, starting with $x_0 = 0$, $y_0 = 0$, we have for the first step:

$$f(x_0, y_0) = x_0 + y_0 = 0 + 0 = 0$$

and so

$$y_{1,p} = y_0 + hf(x_0, y_0) = 0 + (.5)(0) = 0$$

But now $x_1 = 0.5$, and so $f(x_1, y_{1,p}) = x_1 + y_{1,p} = (.5) + (0) = .5$
and therefore

$$y_{1,c} = y_0 + \frac{h}{2}\Big(f(x_0, y_0) + f(x_1, y_{1,p})\Big) = (0) + \frac{(.5)}{2}\Big((0) + (.5)\Big) = 0.125$$

solution:

Then, for the second step, we have $f(x_1, y_1) = (.5) + (.125) = .625$ and so

$$y_{2,p} = y_1 + hf(x_1, y_1) = (.125) + (.5)(.625) = .4375$$

and so, since $x_2 = 1.0$, $f(x_2, y_{2,p}) = (1.0) + (.4375) = 1.4375$

and therefore

$$y_{2,c} = y_1 + \frac{h}{2}\Big(f(x_1, y_1) + f(x_2, y_{2,p})\Big)$$

$$= (.125) + \frac{(.5)}{2}\Big((.625) + (1.4375)\Big) = 0.6406$$

i.e.

$$y(1) \doteq 0.6406$$

b. Does the behavior of the error at $x = 1.$ in this example reasonably agree with theory? (*Briefly* explain your answer!)

solution:

Consider the following table:

h	y_{true}	y_{approx}	error
.25	0.7183	0.6949	0.0234
.50	0.7183	0.6406	0.0777

$$\Longrightarrow \quad \frac{\text{error}(h = .5)}{\text{error}(h = .25)} = \frac{.0777}{.0234} \doteq 3.32$$

This method has second order global error, and therefore the errors should be approximately proportional to h^2. If the error were exactly proportional to h^2 we would have a ratio exactly equal to four. But this isn't that far off, so the result agrees reasonably with theory.

c. Using the above *table*, estimate the value of $y(1.25)$ using the second-order Adams-Bashforth-Moulton method:

$$y_{n+1,p} = y_n + \frac{h}{2}\,(3\,f_n \;-\; f_{n-1})$$
$$y_{n+1,c} = y_n + \frac{h}{2}\,(f_{n+1,p} \;+\; f_n)$$

solution:

First, note that $n = 4$, and so

$$f_4 = f(x_4, y_4) = f(1.00, 0.6949) = 1.00 + 0.6949 = 1.6949$$

Then, using the predictor formula

$$y_{5,p} = y_4 + \frac{h}{2}\,(3f_4 - f_3) = 0.6949 + \frac{.25}{2}\;(3(1.6949) - (1.1033)) = 1.1926$$

Now,

$$f_{5,p} = f(x_5, y_{5,p} = f(1.25, 1.1926) = 2.4426$$

and so

$$y_{5,c} = y_4 + \frac{h}{2}\,(f_{5,p} + f_4) = 0.6949 + \frac{.25}{2}\;(2.4426 + 1.6949) = 1.2121$$

Hence, $y_5 = 1.2121$

6. (25 points) a. Consider the mathematically equivalent statements:

$$\tilde{f}(x) = x - \sqrt{x^2 - 1} \qquad \text{and} \qquad \hat{f}(x) = \frac{1}{x + \sqrt{x^2 - 1}}$$

Use four-digit decimal arithmetic with rounding of all intermediate results to evaluate both of these expressions numerically at

$$x = 15.0 = .1500 \times 10^2$$

What is the relative error of each of the two expressions? Computationally, which is the preferable form for $f(x)$ and why? (*Briefly* justify your answer.)

solution:

First, note that in four digit arithmetic

$$x^2 - 1 = 15.00^2 - 1.000 = 225.0 - 1.000 = 224.0$$

can be computed exactly. However, with rounding of itermediate results,

$$\sqrt{x^2 - 1} = \sqrt{224.0} = 14.97$$

Hence, in our four digit machine: $\quad x - \sqrt{x^2 - 1} = 15.00 - 14.97 = 0.03000$

and: $\quad x + \sqrt{x^2 - 1} = 15.00 + 14.97 = 29.97$

Finally, after rounding, in our four-digit machine

$$\tilde{f}(x) = x - \sqrt{x^2 - 1} = 0.03000 \quad \text{and} \quad \hat{f}(x) = \frac{1}{x + \sqrt{x^2 - 1}} = 0.03337$$

Since the exact value is $f(x) = 0.03337045...$, then the relative errors are, respectively

$$\frac{f(x) - \tilde{f}(x)}{f(x)} \doteq \frac{0.03337045 - 0.03000}{0.03337045} \doteq 0.101 \doteq 200\ \epsilon_{machine}$$

and

$$\frac{f(x) - \hat{f}(x)}{f(x)} \doteq \frac{0.03337045 - 0.03337}{0.03337045} \doteq 1.36 \times 10^{-5} \doteq 0.027\ \epsilon_{machine}$$

Clearly, the second expression is far superior, and the reason should be obvious. It avoids a subtraction of two nearly equal number, and hence avoids a (near) catastrophic cancellation.

b. A certain approximation for a definite integral is known to have global errors that are $\mathbf{O}(h^3)$. A report concludes that using this method in a problem with a step size of $h = 0.1$ should produce an error on the order of magnitude of $0.001 = (0.1)^3$. What, if anything, is wrong with this conclusion.

solution:

By definition,

$$Error = \mathbf{O}\left(h^3\right) \qquad \Longrightarrow \qquad Error \doteq Ch^3$$

for small h. Therefore, for $h = 0.1$ (which should be small, assuming we've properly scaled the problem),

$$Error \doteq C(0.1)^3 = 0.001\ C$$

However, without any knowledge of the order of magnitude of C, it is impossible to predict the order of magnitude of

$$0.001\ C$$

Therefore the conclusion in the report is wrong! (Although note it should be possible to estimate the order of magnitude of the error *correctly* by either recalculating this integral with a different step size and using extrapolation, or by comparing the answer from this method to that from a higher order method, e.g. Simpson's rule.)